Robust optimization for a maritime inventory routing problem ☆

Agostinho Agra[a], Marielle Christiansen[b], Lars Magnus Hvattum[c], Filipe Rodrigues[a]

[a] *Department of Mathematics and Center for Research and Development in Mathematics and Applications, University of Aveiro, Portugal*
[b] *Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Norway*
[c] *Faculty of Logistics, Molde University College, Norway*

Abstract

We consider a single product maritime inventory routing problem in which the production and consumption rates are constant over the planning horizon. The problem involves a heterogeneous fleet and multiple production and consumption ports with limited storage capacity.

Maritime transportation is characterized by high levels of uncertainty, and sailing times can be severely influenced by varying and unpredictable weather conditions. To deal with the uncertainty, this paper investigates the use of adaptable robust optimization where the sailing times are assumed to belong to the well-known budget polytope uncertainty set.

In the recourse model, the routing, the order of port visits, and the quantities to load and unload are fixed before the uncertainty is revealed, while the visit time to ports and the stock levels can be adjusted to the scenario. We propose a decomposition algorithm that iterates between a master problem that considers a subset of scenarios and an adversarial separation problem that searches for scenarios that make the solution from the master problem infeasible. Several improvement strategies are proposed aiming at reducing the running time of the master problem and reducing the number of iterations of the decomposition algorithm. An iterated local search heuristic is also introduced to improve the decomposition algorithm. A computational study is reported based on a set of real instances.

Keywords: Robust optimization; Uncertainty; Travel time; Decomposition; Matheuristic

☆ Published in Transportation Science, 52, 509–525, 2018.

Email addresses: aagra@ua.pt (Agostinho Agra), mc@iot.ntnu.no (Marielle Christiansen), hvattum@himolde.no (Lars Magnus Hvattum), fmgrodrigues@ua.pt (Filipe Rodrigues)

1. Introduction

In maritime transportation, large cargo quantities are transported between ports. Often storages are placed at or close to the ports at both ends of a sailing leg. The transportation at sea as well as the storages at ports are most often parts of a supply chain from producers to end customers. When a single decision maker has the responsibility for both the transportation of the cargoes and the inventories at the ports, the routing and scheduling of the ships and the inventory management can be planned simultaneously. The resulting problem is called a maritime inventory routing problem (MIRP).

The shipping industry is capital intensive with high investment and operating costs for the ships as well as large and valuable cargoes, so a modest improvement in the fleet utilization can imply a large increase in profit. Therefore, the MIRP is a very important and common problem in maritime shipping. These reasons, as well as the difficulties to solve the problem due to high degree of freedom in the routing, scheduling, number of port visits, and the loaded and unloaded quantity, has lead to a solid amount of research on MIRPs. The resulting publications have formed the basis of several surveys: Papageorgiou et al. [39], Christiansen et al. [27], and Christiansen and Fagerholt [25, 26]. In addition, Coelho et al. [29] and Andersson et al. [10] surveyed both land-based and maritime inventory routing problems.

Maritime transportation is characterized by high levels of uncertainty, and one of the most prevalent sources of uncertainty is the sailing times that are affected heavily by changing weather conditions. In practice, unpredictable delays may affect the execution of an otherwise optimal deterministic plan. In order to compensate for such delays, it is possible for the ships to speed up when necessary. However, in practice it will most often be beneficial to consider the uncertainty explicitly when finding the optimal plan.

Therefore, we consider a maritime inventory routing problem (MIRP) with uncertain sailing or travelling times. A heterogeneous fleet of ships is transporting a single product between ports. There is one set of ports where the product is produced, and another set of ports where the product is consumed. The production and consumption rates are assumed constant over the planning horizon. In all ports, there exists a storage for the product, and lower and upper inventory limits are given for each storage. Each port can be visited once or several times during the planning horizon depending on the size of the storage, the production or consumption rate, and the quantity loaded or unloaded at each port visit. The MIRP with uncertain travelling times consists of designing routes and schedules for a fleet of ships that are robust against delay in travelling times in order to minimize the transportation and port costs, and to determine the quantities handled at each port call without exceeding the storage capacities.

Even though maritime transportation is heavily influenced by uncertainty, most of the research reported in the literature on maritime routing and scheduling consider static and deterministic problems. We review some of the existing contributions within maritime transportation considering uncertainties.

For a ship routing and scheduling problem with predefined cargoes, Christiansen and

Fagerholt [24] design ship schedules that are less likely to result in ships staying idle at ports during weekends by imposing penalty costs for arrivals at risky times (i.e. close to weekends). The resulting schedule needs to be more robust with respect to delays from bad weather and unpredictable time in port due to the restricted operating hours each day and ports being closed during weekends. Agra et al. [7] solved a full-load ship routing and scheduling problem with uncertain travel times using robust optimization. Furthermore, Halvorsen-Weare and Fagerholt [34] analysed various heuristic strategies to achieve robust weekly voyages and schedules for off-shore supply vessels working under tough weather conditions. Heuristic strategies for obtaining robust solutions with uncertain sailing times and production rate were also discussed by Halvorsen-Weare et al. [35] for the delivery of liquefied natural gas.

For a crude oil transportation and inventory problem, Cheng and Duran [23] developed a decision support system that takes into account uncertainty in sailing time and demand. The problem was formulated as a discrete time Markov decision process and solved by using discrete event simulation and optimal control theory. Rakke et al. [40] and Sherali and Al-Yakoob [44, 45] introduced penalty functions for deviating from the customer contracts and the storage limits, respectively, for their MIRPs. Christiansen and Nygreen [28] used soft inventory levels to handle uncertainties in sailing time and time in port, and these levels were transformed into soft time windows for a single product MIRP. Agra et al. [6] were the first to use stochastic programming to model uncertain sailing and port times for a MIRP with several products and inventory management at the consumption ports only. Recently, a heuristic stochastic approach is presented in Agra et al. [8] to be able solve larger instances of the MIRP. Additionally, the authors explain why using penalties for backlogged demands make the deterministic problem much harder, which also motivates the recourse to robust approaches for MIRP.

Zhang et al. [51], see also Zhang [50], developed robust approaches for an Annual Delivery Plan problem involving a single producer and multiple customers in the Liquefied Natural Gas business. First, a maritime inventory routing problem with given time windows for deliveries with uncertain travel disruptions is solved by use of a Lagrangian heuristic scheme to obtain robust solutions. Second, a more general robust maritime inventory routing problem with time windows is studied, where the length and placement of the time windows are also decision variables. The problem is formulated as a two-stage stochastic mixed-integer program, and the author proposes a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions.

Robust inventory routing problems has also been considered in land transportation, but the uncertainty is related to the demands. Solyali et al. [46] proposed a dualization approach, while Agra et al. [9] developed a decomposition approach for inventory models that can be combined with routing and uncertain demands. More general robust inventory problems have been considered, see for instance [19], however the inventory problems usually assume the time is discretized into a finite set of time periods, which contrasts with our problem where the time is considered continuous. For recent overviews on robust optimization see [12, 14, 22, 33].

Uncertainty has been considered for other related routing problems recently. Roldán et al. [41] present three new customer selection methods for a dynamic and stochastic inventory routing problem. Aghezzaf [3] considers a variant of the inventory routing optimization problem where customer demand rates and travel times are stochastic but stationary. He proposes an approach to generate optimal robust distribution plans. Li et al. [36] consider an inventory routing problem under replenishment lead-time where the inventory levels are uncertain. They propose and optimization approach based on a genetic algorithm. Bertsimas et al. [15] introduce a scalable approach for solving a robust and adaptive mixed integer formulation for an inventory routing problem with demand uncertainty. Desaulniers et al. [30] consider a new mathematical formulation for the IRP and develop a state-of-the-art branch-price-and-cut algorithm for solving the problem. A survey on the inventory routing problem with stochastic lead times and demands can be found in [42].

Zhang et al. [52] study a robust maritime inventory routing problem with time windows and stochastic travel times, where the length and placement of the time windows are decision variables. The problem is modeled as a two-stage stochastic mixed-integer program, and a two-phase heuristic solution approach is proposed.

Adulyasak and Jaillet [1] consider the vehicle routing problem with deadlines under travel time uncertainty, discussing both stochastic and robust problem variants. Adulyasak et al. [2] solve the land production routing problem under demand uncertainty considering a stochastic setting by using a Benders decomposition approach with several enhancements. Other heuristic algorithms for the production routing problem were proposed by Solyali and Süral [47] and Russell [43]. A comparison between two scenario-based frameworks, a stochastic programming and robust optimization approach, for supply planning under uncertainty is provided by Maggioni et al. [38].

The objective of this paper is to present a general robust optimization procedure for solving single product MIRP with uncertain travelling times that results in robust solutions that are immune to some sailing times delays and where the inventory limits are not violated due to the delays. In the robust model, the travelling times belong to an uncertainty set, which we assume to be the well known budget constrained polytope introduced by Bertsimas and Sim [17]. The total deviation of the travelling times to the nominal values is controlled by a parameter. This set has the advantage that it is easy to interpret from a practical point of view, and its structure can be explored from a computational point of view when decomposition techniques are employed [7, 9, 20].

In relation to existing literature, this paper provides the following contributions:

(i) introduces a robust model to a MIRP in order to derive solutions that are immune to a certain number of delays in relation to inventory level deviations. This model assumes that the routing, number of port visits and the quantities to load and unload cannot be adjusted to the uncertain scenario, while the time for start of service as well as the inventory levels are adjustable;

(ii) develops a decomposition algorithm, where the problem is relaxed into a master

problem and each robust constraint is written for a small subset of scenarios only, and a separation subproblem that checks whether the solution is feasible for the omitted robust constraints;

(iii) introduces several improvement strategies for the decomposition algorithm. One set of improvements aims to reduce the running time of each master problem, while the other intends to reduce the number of iterations of the decomposition algorithm. Most of these improvements can be extended to other related problems solved by robust optimization;

(iv) a new iterated local search heuristic is presented. The heuristic provides good quality solution and can also be extended to other robust optimization problems. The heuristic is used to improve the exact decomposition approach.

The rest of the paper is organized as follows: The mathematical model of the deterministic problem is presented in Section 2, while the robust optimization model and the decomposition algorithm are described in Section 3. Section 4 is devoted to improvement strategies for the decomposition approach. An iterated local search heuristic is presented in Section 5. Furthermore, computational results are reported and discussed in Section 6, followed by some concluding remarks in Section 7.

2. Mathematical model for the deterministic problem

In this section we present a mathematical formulation for a deterministic version of our maritime inventory routing problem.

Routing constraints

Let V denote the set of ships and N denote the set of ports. Each ship $v \in V$ must depart from its initial position, which is either a port or a point at sea. For each port we consider an ordering of the visits accordingly to the time of the visit.

The ship paths are defined on a network where the nodes are represented by a pair (i, m), where i indicates the port and m indicates the visit number to port i. Direct ship sailings (arcs) from node (i, m) to node (j, n) are represented by (i, m, j, n). Figure 1 depicts two ship paths. Ship 1 leaves its origin, sails to Port 1, for the first visit, then Port 2 is visited for the first time, and finally ship 1 terminates its route servicing Port 3. This is the second visit to Port 3, because Ship 2 visited Port 3 first on its route to Port 1.

We define S^A as the set of possible nodes (i, m), S_v^A as the set of nodes that may be visited by ship v, and set S_v^X as the set of all possible sailings (i, m, j, n) of ship v. For the routing we define the following binary variables: x_{imjnv} is 1 if ship v travels from node (i, m) directly to node (j, n), and 0 otherwise; x_{imv}^O indicates whether ship v travels directly from its initial position to node (i, m) or not; w_{imv} is 1 if ship v visits node (i, m), and 0 otherwise; z_{imv} is equal to 1 if ship v ends its route at node (i, m), and 0

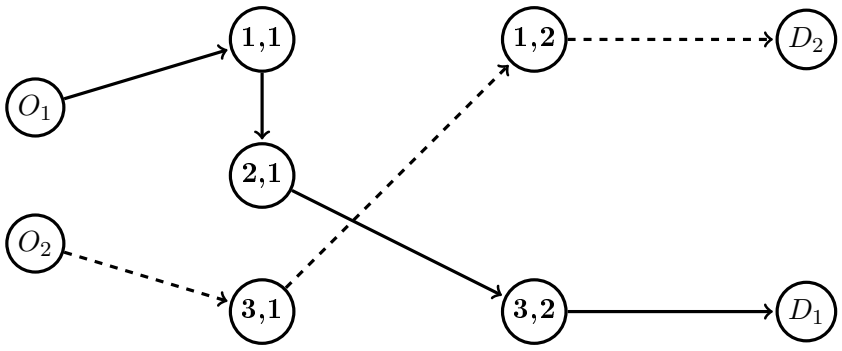

Figure 1: Example of two ship routes. The route of Ship 1 is represented by solid lines and the route of Ship 2 is represented by dashed lines.

otherwise; z_v^O is equal to 1 if ship v is not used and 0 otherwise; y_{im} indicates whether a ship is making the m^{th} visit to port i, (i,m), or not. The parameter $\underline{\mu}_i$ denotes the minimum number of visits at port i and the parameter $\overline{\mu}_i$ denotes an upper bound on the number of visits at port i.

$$\sum_{(i,m)\in S_v^A} x_{imv}^O + z_v^O = 1, \qquad \forall v \in V, \tag{1}$$

$$w_{imv} - \sum_{(j,n)\in S_v^A} x_{jnimv} = 0, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{2}$$

$$w_{imv} - \sum_{(j,n)\in S_v^A} x_{imjnv} - z_{imv} = 0, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{3}$$

$$\sum_{v\in V} w_{imv} = y_{im}, \qquad \forall (i,m) \in S^A, \tag{4}$$

$$y_{im} = 1, \qquad \forall (i,m) \in S^A : m \in \{1,\ldots,\underline{\mu}_i\}, \tag{5}$$

$$y_{i(m-1)} - y_{im} \geq 0, \qquad \forall (i,m) \in S^A : \underline{\mu}_i + 1 < m \leq \overline{\mu}_i, \tag{6}$$

$$x_{imjnv} \in \{0,1\}, \qquad \forall v \in V, (i,m,j,n) \in S_v^X, \tag{7}$$

$$x_{imv}^O, w_{imv}, z_{imv} \in \{0,1\}, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{8}$$

$$y_{im} \in \{0,1\}, \qquad \forall (i,m) \in S^A. \tag{9}$$

Equations (1) ensure that each ship either departs from its initial position and travels to another node or the ship is not used. Equations (2) and (3) are the flow conservation constraints, ensuring that a ship arriving at a node either leaves that node or ends its route. Constraints (4) ensure that a ship can visit node (i,m) only if y_{im} is equal to one. Equations (5) fix y_{im} to 1 for the mandatory visits. Constraints (6) state that if port i is visited m times, then it must also have been visited $m-1$ times. Constraints (7)-(9) define the variables as binary.

Loading and unloading constraints

Parameter J_i is 1 if port i is a producer and -1 if it is a consumer. The quantity on ship v at the beginning of the planning horizon is given by Q_v^O, and the capacity of ship v is denoted by C_v. The minimum and maximum loading and unloading quantities at port i are given by $\underline{Q}_i$ and $\overline{Q}_i$, respectively.

In order to model the loading and unloading constraints, we define the following continuous variables: q_{imv} is the amount loaded or unloaded from ship v at node (i,m); f_{imjnv} denotes the amount that ship v transports from node (i,m) to node (j,n), and f_{imv}^O gives the amount that ship v transports from its initial position to node (i,m). The loading and unloading constraints are given by:

$$f_{imv}^O + \sum_{(j,n)\in S_v^A} f_{jnimv} + J_i q_{imv} = \sum_{(j,n)\in S_v^A} f_{imjnv}, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{10}$$

$$f_{imv}^O = Q_v^O x_{imv}^O, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{11}$$

$$f_{imjnv} \leq C_v x_{imjnv}, \qquad \forall\ v \in V, (i,m,j,n) \in S_v^X, \tag{12}$$

$$\underline{Q}_i w_{imv} \leq q_{imv} \leq min\{C_v, \overline{Q}_i\} w_{imv}, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{13}$$

$$f_{imjnv} \geq 0, \qquad \forall v \in V, (i,m,j,n) \in S_v^X, \tag{14}$$

$$f_{imv}^O, q_{imv} \geq 0, \qquad \forall v \in V, (i,m) \in S_v^A. \tag{15}$$

Equations (10) are the flow conservation constraints at node (i,m). Equations (11) determine the quantity on ship v when it travels from its initial node to node (i,m). Constraints (12) require that the ship capacity is obeyed. Constraints (13) impose lower and upper limits on the loading and unloading quantities. Constraints (14)-(15) are the non-negativity constraints.

Time constraints

We define the following parameters: T_i^Q is the time required to load or unload one unit of product at port i and T_{ijv} is the travel time between port i and j by ship v. The travel time also includes any set-up time required to operate at port j. T_{iv}^O indicates the travelling time required by ship v to travel from its initial position to facility i. T_i^B is the minimum time between two consecutive visits to port i. T is the length of the time horizon, and A_{im} and B_{im} are the time windows for starting the m^{th} visit to port i. Such time windows are considered only for generality, since here we will consider $A_{im} = 0$, and $B_{im} = T$. To ease the presentation we also define, for each node (i,m), the following upper bound for the end time of the visit: $T'_{im} = \min\{T, B_{im} + T_i^Q \overline{Q}_i\}$. Given time variables t_{im} that indicate the start time of each visit at each port, the time constraints

can be written as:

$$t_{im} + \sum_{v \in V:(i,m,j,n) \in S_v^X} \max\{T'_{im} + T_{ijv} - A_{jn}, 0\} x_{imjnv} + \sum_{v \in V} T_i^Q q_{imv} - t_{jn} \leq T'_{im} - A_{jn}, \qquad \forall (i,m),(j,n) \in S^A, \quad (16)$$

$$t_{im} - t_{i,m-1} - \sum_{v \in V} T_i^Q q_{i,m-1,v} - T_i^B y_{im} \geq 0, \qquad \forall (i,m) \in S_A : m > 1, \quad (17)$$

$$\sum_{v \in V} T_{iv}^O x_{imv}^O \leq t_{im}, \qquad \forall (i,m) \in S^A, \quad (18)$$

$$A_{im} \leq t_{im} \leq B_{im}, \qquad \forall (i,m) \in S^A. \quad (19)$$

Constraints (16) relate the start time associated with node (i,m) to the start time associated with node (j,n) when ship v travels directly from (i,m) to (j,n). Constraints (17) impose a minimum interval between two consecutive visits at port i. Constraints (18) ensure that if ship v travels from its initial position to (i,m), then the start time associated with (i,m) is at least the travelling time between the initial position and port i. Time windows for the start time of visits are given by constraints (19).

Inventory constraints

The inventory constraints are considered for each port. They ensure that the stock levels are within the corresponding limits and link the stock levels to the loading or unloading quantities. For each port i, the rate of consumption or production, R_i, the minimum $\underline{S}_i$, the maximum $\overline{S}_i$ and the initial S_i^0 stock levels are given. We define the nonnegative continuous variables s_{im} to represent the stock levels at the start of the m^{th} visit to port i. The inventory constraints are as follows:

$$s_{i1} = S_i^0 + J_i R_i t_{i1}, \qquad \forall i \in N, \quad (20)$$

$$s_{im} = s_{i,m-1} - J_i \sum_{v \in V} q_{i,m-1,v} + J_i R_i (t_{im} - t_{i,m-1}), \qquad \forall (i,m) \in S^A : m > 1, \quad (21)$$

$$s_{im} + \sum_{v \in V} q_{imv} - R_i \sum_{v \in V} T_i^Q q_{imv} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \quad (22)$$

$$s_{im} - \sum_{v \in V} q_{imv} + R_i \sum_{v \in V} T_i^Q q_{imv} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = 1, \quad (23)$$

$$s_{i\overline{\mu}_i} + \sum_{v \in V} q_{i,\overline{\mu}_i,v} - R_i (T - t_{i\overline{\mu}_i}) \geq \underline{S}_i, \qquad \forall i \in N : J_i = -1, \quad (24)$$

$$s_{i\overline{\mu}_i} - \sum_{v \in V} q_{i,\overline{\mu}_i,v} + R_i (T - t_{i\overline{\mu}_i}) \leq \overline{S}_i, \qquad \forall i \in N : J_i = 1, \quad (25)$$

$$s_{im} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \quad (26)$$

$$s_{im} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A : J_i = 1. \quad (27)$$

Equations (20) calculate the stock level at the start time of the first visit to a port, and equations (21) relate the stock level at the start time of the m^{th} visit to the stock level at the start time of the previous visit. Constraints (22) and (23) ensure that the stock levels are within their limits at the end of each visit. Constrains (24) impose a lower bound on the inventory level at time T for consumption ports, while constrains (25) impose an upper bound on the inventory level at time T for production ports. Constraints (26) and (27) ensure that the stock levels are within their limits at the start of each visit.

Objective function

The objective is to minimize the total routing costs, including travelling and operating costs. The travelling cost of ship v from port i to port j is denoted by C^T_{ijv} and it includes the set-up costs. C^{TO}_{iv} represents the travelling cost of ship v from its initial position to port i. The objective function is defined as follows:

$$\min \qquad C(X) = \sum_{v \in V} \sum_{(i,m,j,n) \in S^X_v} C^T_{ijv} x_{imjnv} + \sum_{v \in V} \sum_{(i,m) \in S^A_v} C^{TO}_{iv} x^O_{imv}. \tag{28}$$

3. Robust optimization

In this section we first present the robust optimization model and then describe the solution method proposed.

3.1. Mathematical model for the robust formulation

In the robust model the travelling times belong to an uncertainty set. The uncertainty set represents the situation where there can be at most a number Γ of delays in the ship paths. Instead of the travelling time T_{ijv}, we might add a delay $\hat{T}_{ijv}$ to a nominal travelling time value $\bar{T}_{ijv}$. As the travelling times do not depend on the visits, one would increase all the travelling times between i and j if T_{ijv} is increased. This increase affects mostly those routes where a ship sails multiple times directly between the same two ports. To have full control on the number of delays, we replace T_{ijv} by ξ_{imjnv} in constraints (16) and T^O_{iv} by ξ^O_{imv} in constraints (18):

$$t_{im} + \sum_{v \in V | (i,m,j,n) \in S^X_v} \max\{T'_{im} + \xi_{imjnv} - A_{jn}, 0\} x_{imjnv}$$
$$+ \sum_{v \in V} T^Q_i q_{imv} - t_{jn} \leq T'_{im} - A_{jn}, \qquad \forall (i,m),(j,n) \in S^A, \tag{29}$$
$$\sum_{v \in V} \xi^O_{imv} x^O_{imv} \leq t_{im}, \qquad \forall (i,m) \in S^A. \tag{30}$$

For ease of notation we will consider in the following T^O_{imv} as a particular case of T_{jnimv} where j is the initial position of ship v, denoted as $o(v)$, and n will be 1. The uncertainty

set is now defined using the travelling times that depend on the visits, as follows:

$$\Xi^{\Gamma} = \{\xi : \xi_{imjnv} = \bar{T}_{ijv} + \hat{T}_{ijv}\delta_{imjnv}, \\ 0 \leq \delta_{injmv} \leq 1, v \in V, (i,m,j,n) \in S_v^X, \sum_{v \in V} \sum_{(i,m,j,n) \in S_v^X} \delta_{imjnv} \leq \Gamma\}.$$

This uncertainty set is the well-known budget polytope introduced by Bertsimas and Sim [18], where $\bar{T}_{ijv}$ is the nominal value corresponding to the expected travel time, $\hat{T}_{ijv}$ is the maximum allowed deviation (delay), δ_{injmv} is the deviation of parameter T_{imjnv} from its nominal value, and Γ limits the number of deviations.

The model introduced here is an adjustable robust program [21, 13, 31, 48] which features two levels of decisions: first-stage variables must be fixed before the uncertainty is revealed, while adjustable variables can react to account for the uncertainty. Such a concept has also been known as recoverable robustness [37].

The first-stage variables $x_{imjnv}, z_{imv}, w_{imv}, y_{im}$, and q_{imv} are the routing, the port visits sequence, and the loading and unloading decisions. The adjustable variables are those related to the time and the stock levels. These variables now depend upon the uncertain parameters. Hence, we define $t_{im}(\xi)$, and $s_{im}(\xi)$ as the time and the stock level of visit (i,m), respectively, when scenario ξ (vector of travel times) is revealed.

The first stage solution must ensure that, for each possible vector of travel times in the budget polytope, the stock level at each port i is within the inventory bounds $\underline{S}_i$ and $\overline{S}_i$. For the robust model the time and inventory constraints are replaced by the following constraints:

Time constraints:

$$t_{im}(\xi) + \sum_{v \in V:(i,m,j,n) \in S_v^X} \max\{T'_{im} + \xi_{imjnv} - A_{jn}, 0\} x_{imjnv}$$

$$-t_{jn}(\xi) + \sum_{v \in V} T_i^Q q_{imv} \leq T'_{im} - A_{jn}, \qquad \forall (i,m,j,n) \in S^X, \xi \in \Xi^{\Gamma}, \tag{31}$$

$$t_{im}(\xi) - t_{i,m-1}(\xi) - \sum_{v \in V} T_i^Q q_{i,m-1,v} - T_i^B y_{im} \geq 0, \forall (i,m) \in S_A : m > 1, \xi \in \Xi^{\Gamma}, \tag{32}$$

$$\sum_{v \in V} \xi_{o(v)1imv} x^O_{imv} \leq t_{im}(\xi), \qquad \forall (i,m) \in S^A, \xi \in \Xi^{\Gamma}, \tag{33}$$

$$A_{im} \leq t_{im}(\xi) \leq B_{im}, \qquad \forall (i,m) \in S^A, \xi \in \Xi^{\Gamma}. \tag{34}$$

Inventory management constraints:

$$s_{i1}(\xi) = S_i^0 + J_i R_i t_{i1}(\xi), \qquad \forall i \in N, \xi \in \Xi^\Gamma, \tag{35}$$

$$s_{im}(\xi) = s_{i,m-1}(\xi) - J_i \sum_{v \in V} q_{i,m-1,v} + J_i R_i (t_{im}(\xi) - t_{i,m-1}(\xi)), \qquad \forall (i,m) \in S^A : m > 1, \xi \in \Xi^\Gamma, \tag{36}$$

$$s_{im}(\xi) + \sum_{v \in V} q_{imv} - R_i \sum_{v \in V} T_i^Q q_{imv} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \xi \in \Xi^\Gamma, \tag{37}$$

$$s_{im}(\xi) - \sum_{v \in V} q_{imv} + R_i \sum_{v \in V} T_i^Q q_{imv} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = 1, \xi \in \Xi^\Gamma, \tag{38}$$

$$s_{i\overline{\mu}_i}(\xi) + \sum_{v \in V} q_{i,\overline{\mu}_i,v} - R_i(T - t_{i\overline{\mu}_i}(\xi)) \geq \underline{S}_i, \qquad \forall i \in N : J_i = -1, \xi \in \Xi^\Gamma, \tag{39}$$

$$s_{i\overline{\mu}_i}(\xi) - \sum_{v \in V} q_{i,\overline{\mu}_i,v} + R_i(T - t_{i\overline{\mu}_i}(\xi)) \leq \overline{S}_i, \qquad \forall i \in N : J_i = 1, \xi \in \Xi^\Gamma, \tag{40}$$

$$s_{im}(\xi) \geq \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \xi \in \Xi^\Gamma, \tag{41}$$

$$s_{im}(\xi) \leq \overline{S}_i, \qquad \forall (i,m) \in S^A : J_i = 1, \xi \in \Xi^\Gamma. \tag{42}$$

The robust model is defined by (1)–(15), (28), (31)–(42).

3.2. Model analysis

The model has an infinite number of variables $t_{im}(\xi)$ and $s_{im}(\xi)$, as well as time constraints and inventory constraints. However, as the recourse model (31)–(42) is a pure linear model with no binary variables, similarly to Lemma 1 in [7], it is easy to show that the uncertainty set can be restricted to the extreme points of the budget polytope. That is $\xi \in ext(\Xi^\Gamma)$ in constraints (31)–(42), where $ext(\Xi^\Gamma)$ is the set of extreme points of Ξ^Γ.

For a given set Θ of scenarios the robust model will be denoted by $R - IR(\Theta)$. Hence, the finite model restricted to the set of extreme points $ext(\Xi^\Gamma)$ will be denoted by $R - IR(ext(\Xi^\Gamma))$.

In order to frame our work within robust optimization we make the following remarks:

Remark 1. *(a) The model $R - IR(\Theta)$ has fixed recourse since the coefficients of the second stage variables in the objective function and constraints are deterministic. Every variable is either non-adjustable or fully adjustable, and the uncertainty set can be considered as a scenario-generated uncertainty set.*

(b) The deterministic problem is NP-hard, and it is quite complex since it generalizes and combines NP-hard problems such as the vehicle routing problem with time windows (VRPTW) and inventory problems. Current research is being conducted on that (deterministic) problem, such as valid inequalities and extended formulations. As the focus is on the robust approaches, we will restrict ourselves to established

results to the deterministic case and focus only on small/medium size instances that can be solved to optimality. Remarks will be done on how the results can be extended to larger instances.

(c) *The value of the second stage (fully adjustable) variables do not affect directly the objective function value. That is, for given routes and pick-up and delivery quantities (first stage decisions) all the robust feasible solutions lead to the same objective function value. This is common to the robust VRPTW problem given in [7] where the second-stage variables (the time of the visit to each node) do not affect the value of the solution. As a result, any policy for the adjustable variables that finds a feasible solution for each scenario when such solution exists, is an optimal policy.*

As we are dealing with fixed recourse, in theory the second stage variables, $t_{im}(\xi)$ and $s_{im}(\xi)$, can be eliminated. That is, the set of feasible solutions can be projected onto the space of the first stage variables. This approach was followed in [11] for a simpler two-stage robust network flow and design problem, and in [7] robust approaches were used for the two dimensional spaces (with first stage and with first and second stage variables) for the robust VRPTW problem, albeit the projection has not been done explicitly. In [11] it was shown that even for simple graph structures the separation of the inequalities resulting from the projection is NP-hard. In [7] a path formulation was used for the formulation in the first stage variables space and the corresponding approach was not considered preferable than the one working in the original space. As in our case, the projection would become much more complex, we opted to eliminate only the stock $s_{im}(\xi)$ variables. Moreover, keeping the time variables $t_{im}(\xi)$ allows us to easily use implicitly the policy of assigning to $t_{im}(\xi)$ the earliest possible time of the visit.

Next we eliminate the $s_{im}(\xi)$ variables. Equations (35) and (36) allow us to write stock variables as functions of the time visits:

$$s_{im}(\xi) = S_i^0 + J_i R_i t_{im}(\xi) - J_i \sum_{v \in V} \sum_{n=1}^{m-1} q_{inv}, \forall (i,m) \in S^A, \xi \in \Xi^\Gamma. \tag{43}$$

Using equations (43), constraints (39) and (40), are converted into the following constraints

$$S_i^0 + \sum_{v \in V} \sum_{m:(i,m) \in S^A} q_{imv} \geq R_i T + \underline{S}_i, \qquad \forall i \in N : J_i = -1, \tag{44}$$

$$S_i^0 + R_i T \leq \sum_{v \in V} \sum_{m:(i,m) \in S^A} q_{imv} + \overline{S}_i, \qquad \forall i \in N : J_i = 1. \tag{45}$$

These constraints depend only on the first-stage decisions (decisions with no recourse).

For consumption ports, constraints (37) and (41) imply, respectively,

$$R_i t_{im}(\xi) \geq \sum_{v \in V} \sum_{n=1}^{m} q_{inv} - \sum_{v \in V} R_i T_i^Q q_{imv} + S_i^0 - \overline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \xi \in \Xi^*, \tag{46}$$

$$R_i t_{im}(\xi) \leq \sum_{v \in V} \sum_{n=1}^{m-1} q_{inv} + S_i^0 - \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = -1, \xi \in \Xi^*. \tag{47}$$

For loading ports, constraints (38) and (42) imply:

$$R_i t_{im}(\xi) \geq \sum_{v \in V} \sum_{n=1}^{m} q_{inv} - \sum_{v \in V} R_i T_i^Q q_{imv} - S_i^0 + \underline{S}_i, \qquad \forall (i,m) \in S^A : J_i = 1, \xi \in \Xi^*, \tag{48}$$

$$R_i t_{im}(\xi) \leq \sum_{v \in V} \sum_{n=1}^{m-1} q_{inv} + \overline{S}_i - S_i^0, \qquad \forall (i,m) \in S^A : J_i = 1, \xi \in \Xi^*. \tag{49}$$

In both cases $\xi \in \Xi^* = \Xi^\Gamma$.

Henceforward we consider the robust model as the model defined by the objective (28), and constraints (1)-(15), (44)-(49). For a given set of scenarios Ξ^*, it will be denoted by $R - IR(ext(\Xi^*))$.

Remark 2. *A common approach to handle adjustable robust problems is to employ approximation techniques such as the well known affine decision rules, see [13, 16]. In our problem the approximation to use is not as clear as in the case where time periods are considered. A reasonable possibility would be to write $t_{im}(\xi)$ as an affine function of the travelling times:*

$$t_{im}(\xi) = \sum_{v \in V} \sum_{(k,\ell,j,n) \in S_v^X} \eta_{k\ell jn}^{im} \xi_{k\ell jnv} + \sum_{v \in V} \eta_{o(v)1im}^{im} \xi_{o(v)1imv} + \eta_0^{im}$$

where the new (non-adjustable) variables are $\eta_{k\ell jn}^{im}$ and $\eta_{o(v)1im}^{im}$.

This approximation has two main drawbacks: (i) it does not take into account other factors that are relevant to define the time of the visits, such as the inventory levels and the time between consecutive visits, thus it may restrict the solution space and deteriorate the quality of the solution; (ii) as we do not have time periods all possible ship paths must be considered leading to a large number of variables. Comparing to the (exact) row-column decomposition approach discussed in the next section we can see that, in general, the number of variables used in the approximation would be larger.

Overall, from Remark 1, (c), optimal policies can be easily derived for the adjustable variables which can be efficiently used from a computational point of view as discussed below. Thus, the approximation decision rules do not seem to help in our case as they may deteriorate the quality of the solution and do not seem to simplify the problem much.

3.3. Solution method

Though finite, the model $R-IR(ext(\Xi^{\Gamma}))$ tends to be very large because the number of extreme points of the uncertainty polytope tends to grow rapidly with the problem size. However, as it happens with many MIP models that are defined through a large number of constraints but where only a few of them need to be included in the model, an efficient solution method can be designed which can be seen as a variant of the Benders' decomposition approach. The main difference is that here both columns and variables are added. Within robust optimization such approaches became popular recently, see [7, 9, 20, 48, 49]. This procedure is also known as the *Adversarial approach* [22].

The decomposition approach works as follows: the problem is relaxed into a master problem (MP), where each robust constraint is written only for a small subset $\Xi^* \subset ext(\Xi^{\Gamma})$. Given a feasible solution to the MP, we check whether the solution is feasible for the omitted robust constraints by solving an adversarial separation problem (ASP). If a scenario leading to infeasibility is found we expand Ξ^* and the corresponding columns and rows are added to the MP and the augmented MP is solved again.

The MP is defined by the objective (28), and constraints (1)-(15), (44)-(48) defined for a subset $\Xi^* \subset ext(\Xi^{\Gamma})$. Given a first-stage solution, the ASP amounts to check whether constrains (46)-(48) are satisfied for all $\xi \in ext(\Xi^{\Gamma})$, and if not, add the corresponding violated constraints. The decomposition procedure is summarized in Algorithm 1.

Algorithm 1 A column-and-row generation approach for the robust problem.

1: Initialize the subset $\Xi^* \subseteq ext(\Xi^{\Gamma}) \leftarrow \{\xi^0\}$ where ξ^0 is the scenario with no delays
2: Solve the restricted $R-IR(\Xi^*)$ problem
3: **while** There is a scenario ξ^* in $ext(\Xi^{\Gamma}) \setminus \Xi^*$, leading to a violation of a constraint (46)–(48) **do**
4: Add ξ^* to Ξ^* (and add the corresponding variables and rows to the model)
5: Solve the new restricted $R-IR(\Xi^*)$ problem
6: **end while**

3.4. Adversarial separation problem

Here we discuss the ASP considered in Step 3 of Algorithm 1. Given a solution that considers the scenarios in Ξ^*, the ASP checks whether this solution is feasible for the remaining scenarios, that is, when the first stage decisions are fixed we check if there is a set of at most Γ delays that leads to a violation of an inventory level (lower or upper bound) constraint.

Assume that the values of the first-stage variables are given by $\overline{x}_{imjnv}$, $\overline{z}_{imv}$, $\overline{w}_{imv}$, $\overline{y}_{im}$, $\overline{q}_{imv}$. We can see that when variables q_{imv} are set to $\overline{q}_{imv}$, the robust constraints (46)–(48) define time-windows $\overline{A}_{im} \leq t_{im} \leq \overline{B}_{im}$ for each visit (i,m).

Inequalities (46)-(47) imply the following time windows for t_{im}, with $(i,m) \in S^A | J_i =$

$-1, \xi \in \Xi^{\Gamma}$,

$$\underbrace{-\sum_{v \in V} T_i^Q \overline{q}_{imv} + \frac{\sum_{v \in V} \sum_{n=1}^{m} \overline{q}_{inv} + S_i^0 - \overline{S}_i}{R_i}}_{\overline{A}_{im}} \leq t_{im}(\xi) \leq \underbrace{\frac{\sum_{v \in V} \sum_{n=1}^{m-1} \overline{q}_{inv} + S_i^0 - \underline{S}_i}{R_i}}_{\overline{B}_{im}},$$

and, inequalities (49)–(48) imply the following time windows for $J_i = 1, \xi \in \Xi^{\Gamma}$,

$$\underbrace{-\sum_{v \in V} T_i^Q \overline{q}_{imv} + \frac{\sum_{v \in V} \sum_{n=1}^{m} \overline{q}_{inv} - S_i^0 + \underline{S}_i}{R_i}}_{\overline{A}_{im}} \leq t_{im}(\xi) \leq \underbrace{\frac{\sum_{v \in V} \sum_{n=1}^{m-1} \overline{q}_{inv} + \overline{S}_i - S_i^0}{R_i}}_{\overline{B}_{im}}.$$

For each port visit, we compute the earliest time of visit and then check whether this time is below the upper time window limit. A recursive approach is followed. Let $\alpha((i,m),\gamma)$ be the earliest arrival time at (i,m) when γ arcs are using their maximum travel time. For $\gamma = 0, \ldots, \Gamma$, the value of $\alpha((i,m),\gamma)$ is given by

$$\alpha((i,m),\gamma) = \max \begin{cases} A_{im} \\ \overline{A}_{im} \\ \alpha((i,m-1),\gamma) + \sum_{v \in V} T_i^Q \overline{q}_{i,m-1,v} + T_i^B, m > 1 \\ \alpha((j,n),\gamma) + T_j^Q \overline{q}_{jnv} + \overline{T}_{jiv} : \quad \overline{x}_{jnimv} = 1 \\ \alpha((j,n),\gamma-1) + T_j^Q \overline{q}_{jnv} + \overline{T}_{jiv} + \hat{T}_{ijv} : \quad \overline{x}_{jnimv} = 1 \end{cases}$$

where $\alpha((i,m),\gamma) = -\infty$ if $\gamma < 0$. A_{im} is the lower time window limit to the visit, and $\overline{A}_{im}$ is the lower limit forced by the stock level, as derived above. The third expression accounts for the case where the earliest visit time results from the constraint imposing a time limit between consecutive visits to the same port (forced by constraints (32)). The fourth and fifth expressions account for the case where the service time results from the visit to another port j (implied by constraints (31)). The fourth expression is for the case where there is no delay in the travelling time, and the fifth expression is for the case where a delay occurs.

The values $\alpha((i,m),\gamma)$ are computed following the order given by the times t_{im} observed in the solution of the MP. This is enough to ensure that $\alpha((i,m),\gamma)$ is computed after the value $\alpha((i,m-1),\gamma)$ and values $\alpha((j,n),\gamma)$, $\alpha((j,n),\gamma-1)$, when $\overline{x}_{jnimv} = 1$ in the MP solution.

The earliest starting time of visit (i,m) is $\overline{\alpha}(i,m) = \max\{\alpha((i,m),\gamma) : \gamma \in \{1,\ldots,\Gamma\}\}$. After computing $\overline{\alpha}(i,m)$ for each port visit (i,m), we need to check whether $\overline{\alpha}(i,m) \leq \min\{B_{im}, \overline{B}_{im}\}$. If there is a violated inequality, that means there is a scenario leading to an inventory limit violation, and the constraints and variables corresponding to that scenario are added. Notice that this separation algorithm generalizes the one given in [7] where only time windows are considered.

Example 1. *Consider the example with* $T = 20$, $V = \{1, 2\}$ *and* $N = \{1, 2, 3\}$. *Port 1 is a supply port and ports 2 and 3 are demand ports. The production/demand rates, the initial stock levels, are given by* $R_1 = 5, R_2 = 1, R_3 = 2$, $S_1^0 = 22, S_2^0 = 10, S_3^0 = 10$, *respectively. Consider* $\underline{S}_i = 0$ *and* $\overline{S}_i = 50$, *for all* $i = 1, 2, 3$. *The quantities loaded/unloaded in the solution are* $q_{111} = 37$, $q_{211} = 10$, $q_{321} = 22$, $q_{312} = 8$, $q_{122} = 45$. *The initial load onboard the ships are* $Q_1^0 = 0$, *and* $Q_2^0 = 8$, *and assume the operation times are negligible* $T_1^Q = T_2^Q = 0$. *The routes are depicted in Figure 2. The values assigned to each arc represent the travelling times. We consider an uncertainty set with* $\Gamma = 2$ *and* $\hat{T}_{ijv} = 1$ *for all* (i, j, v). *The plus 1 next to the values assigned to arcs* $(1, 1, 2, 1)$ *and* $(2, 1, 3, 2)$ *represent the delay of 1 unit. The values* $\alpha((i, n), \gamma)$ *are given for the critical path leading to node* $(3, 2)$. *The value* $\alpha((1, 1), 0)$ *is given by* $\overline{A}_{11}$ *which is computed from the initial stock* $(S_1^0 = 22)$, *the production rate* $R_1 = 5$, *and the load quantity* $(q_{111} = 37)$. *The values of* $\alpha((2, 1), 1)$ *and* $\alpha((3, 2), 2)$ *are computed from the expression* $\alpha((i, m), \gamma) = \alpha((j, n), \gamma - 1) + T_j^Q \overline{q}_{jnv} + \overline{T}_{jiv} + \hat{T}_{ijv}$ *for the routing arcs* $(1, 1, 2, 1)$ *and* $(2, 1, 3, 2)$. *We can see that the solution presented is feasible for the deterministic problem with the nominal travelling times but is infeasible for the robust problem since the earliest arrival time to node* $(3, 2)$ *is 10 leading to a stockout since* $\overline{B}_{32} = 9$.

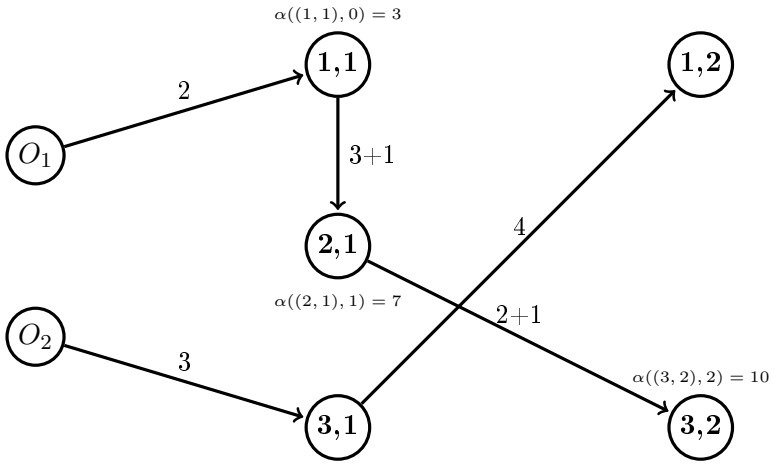

Figure 2: Example of computation of $\alpha((i, m), \gamma)$ for $\Gamma = 2$ for an instance with three ports and two ships.

4. Improvement strategies for the decomposition algorithm

In order to improve the performance of the decomposition approach we test two types of improvements. The first aims to reduce the running time of each master problem, while the second aims to reduce the number of iterations of the decomposition method.

4.1. Solving the master problem

In order to improve the MP solution time we propose two improvements: the inclusion of valid inequalities and the inclusion of a cutoff value. In both cases, these improvements can lead to a significant improvement in the solution times for some difficult instances, while having a negligible influence when solving easy instances.

Valid inequalities

Inequalities from knapsack relaxations have previously been used for MIRPs, see for instance [4, 5]. Here we add a subset of those inequalities that has proved to provide improvements on the integrality gap of related problems.

Let D_i denote the total net demand of port $i \in V$ during the planning horizon. For $i \in N$ with $J_i = -1$, $D_i = T \times R_i - S_i^0 + \underline{S}_i$. For $i \in N$ with $J_i = 1$, $D_i = T \times R_i + S_i^0 - \overline{S}_i$ Then, the following integer set, for $i \in N$, is a relaxation of the feasible set.

$$\Theta_i = \left\{ \chi \in \mathbb{Z}_+^{|V|} : \sum_{v \in V} C_v \chi_v \geq D_i \right\},$$

where

$$\chi_v = \sum_{m:(i,m) \in S_v^A} w_{imv},$$

denotes the number of times vehicle v visits port i during the planning horizon T.

Valid inequalities for $\Theta_i, i \in N$ are valid for the set of feasible solutions. A particular case of these inequalities is the following integer rounding cut

$$\sum_{v \in V} \sum_{m:(i,m) \in S_v^A} \left\lceil \frac{C_v}{Q} \right\rceil w_{imv} \geq \left\lceil \frac{D_i}{Q} \right\rceil, \tag{50}$$

where Q can be any positive number. We consider a different inequality for each $v \in V$, taking $Q = Q_v$.

Initial primal bound

A common upper bound can be obtained by solving the well-known box-constrained problem by considering

$$\Xi^B = \{\xi : \xi_{imjnv} = \bar{T}_{ijv} + \hat{T}_{ijv}\delta_{imjnv}, 0 \leq \delta_{imjnv} \leq 1, \forall (i,m,j,n) \in S^X, v \in V\}.$$

The worst case occurs when all the travelling times take their maximum value, that is, when we consider the deterministic travelling times $T_{imjnv} = \bar{T}_{ijv} + \hat{T}_{ijv}$. Any feasible solution to this deterministic problem is feasible for the robust problem, and therefore, its objective function value provides an upper bound, which can be embedded in the decomposition procedure as a cutoff value to prune the search tree when solving each optimization problem in Step 2 of Algorithm 1. The approach followed in this paper is to solve the box-constrained problem until a desired optimality gap is attained.

4.2. Minimizing the number of iterations of the decomposition approach

The running time of each instance depends greatly on the number of MPs solved. Here we propose three improvement strategies to reduce the number of iterations of the decomposition approach. The first one is aggregation of scenarios. The second improvement strategy facilitates a warmer start considering an initial scenario that leads to an initial solution which incorporates some degree of immunity against delays. Thus the corresponding solution is likely to provide a better lower bound to the value of the optimal solution than the one obtained with the deterministic case where no delays are considered. Finally, we adjust our solution approach to choose values on the first-stage variables that maximizes a given criterion of robustness.

Scenarios aggregation

An improvement introduced in [7] is to add, in each iteration, a scenario that results from the aggregation of several scenarios. Instead of adding a delay on arc (i, m, j, n) for a given ship, one can either add a delay to all arcs (k, ℓ, j, n) that enter node (j, n) or add a delay to all arcs (i, m, k, ℓ) that leave node (i, m). This follows from the fact that at most one such arc can be selected in each feasible solution. Hence, many scenarios can be aggregated into one *augmented scenario*.

Warm start

A warmer start can be obtained by adding delays, that is, consider $T_{imjnv} = \bar{T}_{ijv} + \hat{T}_{ijv}$, for some arcs. These delays need to be carefully added since one need to guarantee that no more than Γ delays are added to any feasible solution. An easy way to ensure that no more than Γ delays are added is either to add delays to all leaving arcs from port visit (i, m), or add delays to all those arcs entering port visit (j, n), for Γ selected port visits. Preferably, the selected visits should be among those that are made in all feasible solutions. Therefore, a natural selection starts by choosing the origins of each ship and then consider the first visit to each port, for up to Γ ships.

Choice of first-stage solution maximizing a robustness criterion

The first-stage variables include both the binary variables $\overline{x}_{imjnv}, \overline{z}_{imv}, \overline{w}_{imv}, \overline{y}_{im}$, and the continuous variables $\overline{q}_{imv}$. In general, for each vector of binary variables one can find alternative values for the continuous variables q_{imv} leading to the same objective function value. Hence, given a binary vector representing the binary decisions, it makes sense to choose among all alternative continuous solutions one that maximizes a given robustness criterion. This results in a pure linear problem when the binary variables are fixed as we explain next.

The ASP checks whether the time windows are consistent by assigning to t_{im} the earliest time of visit value and compare it with an upper bound resulting from the

inventory level. Here, since the load/unload quantities q_{imv} are variables, the upper time-windows limit depends on these variables as follows:

$$B_{im}(q) = \begin{cases} \left(\sum_{v \in V} \sum_{n=1}^{m-1} q_{inv} + \overline{S}_i - S_i^0 \right) / R_i, & J_i = 1 \\ \left(\sum_{v \in V} \sum_{n=1}^{m-1} q_{inv} + S_i^0 - \underline{S}_i \right) / R_i, & J_i = -1 \end{cases}$$

This means that the following constraints must be satisfied:

$$t_{im} \leq B_{im}(q), \forall (i,m) \in S^A. \tag{51}$$

Let

$$d_{im} = B_{im}(q) - t_{im}, \forall (i,m) \in S^A \tag{52}$$

denote the slack in the upper time window constraint that measures the delay that can occur on the time visit (i,m) without having a violation. Intuitively, a feasible solution with large values for slack variables d_{im} will be more immune to delays than a solution with small slack values. In order to chose a first-stage solution, two options are considered. The first option (denoted as Option 1) is to maximize the sum of the slack variables $z = \sum_{(i,m) \in S^A} d_{im}$. The second option (denoted as Option 2) is to maximize the smallest slack d_{im} value, that is, to maximize $z = \underline{d}$ where

$$\underline{d} \leq d_{im}, \forall (i,m) \in S^A. \tag{53}$$

For completeness, Algorithm 2 details the steps that replace Step 2 and Step 4 (solve the restricted $R - IR(\Xi^*)$ problem) in Algorithm 1.

Algorithm 2 A two-stage approach for the master problem $R - IR(\Xi^*)$.

1: Solve the restricted $R - IR(\Xi^*)$ problem
2: Fix the the variables $x_{imjnv}, z_{imv}, w_{imv}, y_{im}$ to their solution values
3: Add nonnegative variables d_{im} (and $\underline{d}$ for Option 2), and constraints (51) and (52) (and constraints (53) for Option 2)
4: Solve the resulting linear problem by maximizing function z

Example 2. *In Example 1 we can see that for the same routing decisions there are alternative values of load and unload quantities. For instance, by decreasing* q_{111} *to 32 and increasing* q_{122} *to 50, the earliest time for visit* $(1,1)$ *is now set to 2,* $(\alpha(1,1),0) = 2$, *which leads to a situation where no shortage occurs at node* $(3,2)$.

Remark 3. *One can see that the first stage solution cannot be complemented with a second stage solution if there is a delay in the route to node* (i,m) *greater than* d_{im}. *A lower bound for the worst case delay can be computed as the sum of the* $\min\{k, \Gamma\}$ *highest values of* $\hat{T}_{ijv}$ *in the route to node* (i,m), *where* k *is the number of arcs in that route.*

5. Iterated local search

As the robust MIRP considered is hard to solve, we present an iterated local search heuristic based on local branching [32]. The idea is to consider as the starting solution the most conservative one, the solution obtained for the box uncertainty set. This choice has three advantages: (i) it guarantees that a feasible solution is found; (ii) the starting solution is obtained solving a deterministic problem and can also be obtained heuristically; (iii) for many instances the value of this solution is not far from the value of the optimal robust solution. Therefore, any improvement obtained will generate good upper bounds for those cases.

For the local search, following the local branching idea of Fischetti and Lodi [32], we define the neighborhood of a solution as the set of solutions that can differ in at most Λ of the w_{imv} variables from the current solution. The local search can be done by adding the following inequality to the model

$$\sum_{(i,m)\in S_v^A, v\in V|\overline{w}_{imv}=0} w_{imv} + \sum_{(i,m)\in S_v^A, v\in V|\overline{w}_{imv}=1} (1 - w_{imv}) \leq \Lambda. \tag{54}$$

The iterated local search is given in Algorithm 3.

Algorithm 3 A MIP iterated local search based approach for a given set of scenarios Ξ^*.

1: Solve the deterministic problem corresponding to the box uncertainty set for α seconds
2: **repeat**
3: Add constraint (54) to the model $R - IR(\Xi^*)$
4: Solve the model for β seconds using the warm start enhancement
5: Update the solution $\overline{w}$
6: **until** No improvement in the objective function is observed

The decomposition approach followed in Algorithm 1 decomposes the problem into a MP and a ASP. While the ASP can be solved efficiently, the MP is NP-hard. Solving the MP to optimality in each iteration of the decomposition method may be too time consuming. Moreover, to prove optimality, only the MP considered at the last iteration must be solved to optimality. In order to use this fact we propose to use Algorithm 3 to solve the MP in each iteration of the decomposition algorithm. Only when no new scenario is found we follow Algorithm 1 to prove optimality. The drawback is that we do not know in advance which is the last iteration and whether a new better solution for the MP with the same set of scenarios will be violated by a new scenario. In that case the iterative process must be resumed. The new enhanced column-and-row generation approach is given in Algorithm 4.

Algorithm 4 The enhanced column-and-row generation approach.

1: Initialize the subset $\Xi^* \subseteq ext(\Xi^\Gamma) \leftarrow \{\xi^*\}$ where ξ^* is a scenario with Γ delays (see Warm start in Section 4)
2: Solve the restricted $R - IR(\Xi^*)$ problem using Algorithm 3
3: Perform Steps 2-4 of Algorithm 2
4: **while** There is a scenario ξ^* in $ext(\Xi^\Gamma) \setminus \Xi^*$, leading to a violation of a constraint (46)–(49) **do**
5: Add ξ^* to Ξ^* (and add the corresponding variables and rows to the model)
6: Solve the new restricted $R - IR(\Xi^*)$ problem using Algorithm 3
7: Perform Steps 2-4 of Algorithm 2
8: **end while**
9: Save the current best solution as incumbent
10: Solve the restricted $R - IR(\Xi^*)$ to optimality using the incumbent solution
11: Perform Steps 2-4 of Algorithm 2
12: **while** There is a scenario ξ^* in $ext(\Xi^\Gamma) \setminus \Xi^*$, leading to a violation of a constraint (46)–(49) **do**
13: Add ξ^* to Ξ^* (and add the corresponding variables and rows to the model)
14: Solve the new restricted $R - IR(\Xi^*)$ using the incumbent solution
15: **end while**

Note that the solution obtained through Steps 1 to 8 is feasible for the model $R - IR(\Omega)$. Hence, the enhanced algorithm discussed in Section 4 corresponds to Steps 9-15 of Algorithm 4 where the warm start is replaced by the best robust solution obtained (resulting from Steps $1 - 8$).

6. Computational tests

This section reports the computational experiments carried out to test the solution approaches for a set of instances of a maritime inventory routing problem with possible delays when travelling between ports. All tests were run on a computer with an Intel Core i5-2410M processor, having a 2.30GHz CPU and 8GB of RAM, using the optimization software Xpress Optimizer Version 21.01.00 with Xpress Mosel Version 3.2.0.

6.1. Instances

The instances are based on those presented in [4]. There are two main differences: one is the computation of the travelling times, which we discuss in detail below, and the other is the production and consumption which we assume here to be constant. The number of ports and ships of each instance is given in the second column of Table 1. Operating and waiting costs are time invariant. The size of the deterministic model is given in the following three columns. A time horizon of 30 days is considered. The set of the 21 instances include both *easy* instances and *difficult* instances where the deterministic problem is already *difficult* to solve.

Table 1: Summary statistics for the 21 instances.

Inst.	(| N |,| V |)	Deterministic Model		
		# Rows	# Col.	# Int. Var.
A1, A2, A3	(4,1)	765	545	273
B1, B2, B3	(3,2)	767	590	302
C1, C2, C3	(4,2)	1214	1042	530
D1, D2, D3	(5,2)	1757	1622	822
E1, E2, E3	(5,2)	1757	1622	822
F1, F2, F3	(4,3)	1663	1539	787
G1, G2, G3	(6,5)	4991	5717	2909

The three instances in each group differ from each other in the initial inventory levels. The nominal travelling times $\bar{T}_{ijv}$ are taken as the travelling times of the instances given in [4]. To define the maximum delays allowed, $\hat{T}_{ijv}$, we chose a constant delay α for each instance. This delay was found by increasing the value of Δ using steps of 0.1 until the deterministic problem with $T_{ijv}(\xi) = \bar{T}_{ijv} + \Delta$ for all (i, j) becomes infeasible. The last Δ leading to a feasible instance was selected. This deterministic case corresponds to the box uncertainty set.

6.2. Summary of the computational results

In this section we report the computational results carried out to test and compare the performance of both the iterated local search (ILS) heuristic and the decomposition algorithm.

6.2.1. Decomposition Algorithm

We start by evaluating the benefits of the use of the improvement strategies in the decomposition algorithm. Table 2 displays the results for the instances G1, G2 and G3, which are the most difficult instances. The first column gives the instance, the second column gives the value Δ and the third column indicates the maximum number of delays, Γ, allowed. The values of Γ range from 0 (corresponding to the deterministic model) to the number of delays leading to the same solution as the one obtained for the box uncertainty set. The optimal objective function value is given in Column *Cost*. The last three pairs of columns give the running time in seconds (Columns *Seconds*), and the number of iterations (Columns *Iter*) of the decomposition algorithm for 1) the case where no improvements are used (Columns *No improvements*), 2) the case where all the improvements are included and Option 1 of minimizing the number of iterations improvement is used in Algorithm 2 (Columns *Improve Opt1*), and 3) the case where all the improvements are included and Option 2 is used (Columns *Improve Opt2*). When $\Gamma = 0$, the improved versions are not run, and a sign "-" is used in the corresponding columns. The case $\Gamma = 0$ corresponds to the deterministic case, and the corresponding cost is therefore a lower bound for the instances with other values of Γ.

Table 2: Computational results (time and number of iterations) for instances G1, G2 and G3.

				No improvements		Improve Opt1		Improve Opt2	
Instance	Δ	Γ	Cost	Seconds	Iter	Seconds	Iter	Seconds	Iter
G1	0.2	0	645.8	975	0	-	-	-	-
		0	560.0	536	0	-	-	-	-
G2	1.6	1	560.0	1791	2	1325	2	1375	2
		2	589.9	13303	6	2818	6	1960	3
		3	589.9	14881	9	4728	5	3010	5
		4	631.4	$>$ 604800	$>$22	123981	13	63734	13
		0	550.7	712	0	-	-	-	-
G3	0.4	1	576.2	20999	7	1907	1	3650	4

Note that the instance G2 with $\Gamma = 4$ could not be solved to optimality without improvements, within one week.

Similar results were obtained for the remaining instances, however, in order to simplify the text, those results were aggregated and are presented in Figures 3 and 4. In these two figures we display the aggregated time and the total number of iterations required by each strategy of the decomposition algorithm to solve all the instances in terms of the different levels of protection Γ, respectively.

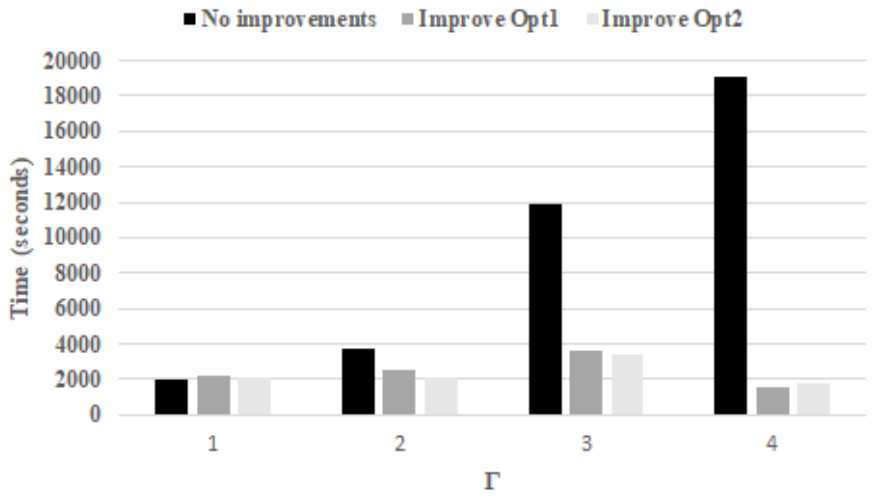

Figure 3: Aggregated computational time for all instances, except for instances G1, G2 and G3, for the different values of Γ

The results show clear benefits of using the improvements in the decomposition algorithm since, when the improvements are employed, the running times can be much lower than in the case where no improvements are used. Instances G2 and G3 are the ones in which more significant improvements can be observed. Note that the benefit of the use

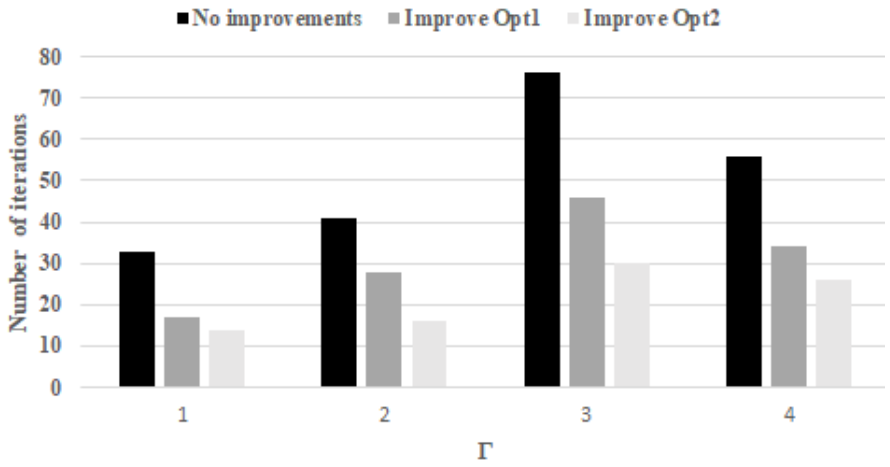

Figure 4: Aggregated number of iterations for all instances, except for instances G1, G2 and G3, for the different values of Γ

of the improvement strategies tends to increase when the level of protection increases. Also the number of iterations is lower when the improvements are included and, in several cases, mainly when Γ is small, no iterations are required. When comparing both improvements options to choose a robust solution among alternative optimal solutions to the master problem, we can see that the number of iterations is lower by using Option 2 than using Option 1. In terms of time there is no clear evidence indicating which is the best improvement option. However, since for the most difficult instance, instance G2, optimal solutions are obtained faster by using the improvement Option 2, in what follows we will only consider the decomposition algorithm enhanced by this option.

6.2.2. Comparison between the decomposition algorithm and the ILS heuristic

For the ILS heuristic, nine different strategies were tested: the time limit in each subproblem β was set to 50, 100 and 200 seconds, and the number of variables w_{imv} that are allowed to flip their value from the value taken in the current solution, parameter Λ, is set to 2, 4 and 5.

For the 21 instances, the solutions obtained by using the ILS heuristic and the decomposition algorithm have the same cost. Hence, the comparison of the solution methods is done only in terms of the running time. For all the instances, except for instances G1, G2 and G3, the results obtained by the nine strategies tested are very similar. This can be explained by the fact that most of the subproblems are solved to optimality within the time limit. Hence, the results for these instances are aggregated and summarized in Figures 5 and 6. In Figure 5, the aggregated computational time required by the decomposition algorithm improved with Option 2 to obtain the robust solution for all the instances, in terms of the values of Γ, is presented. In Figure 6, similar results for

the number of iterations are presented.

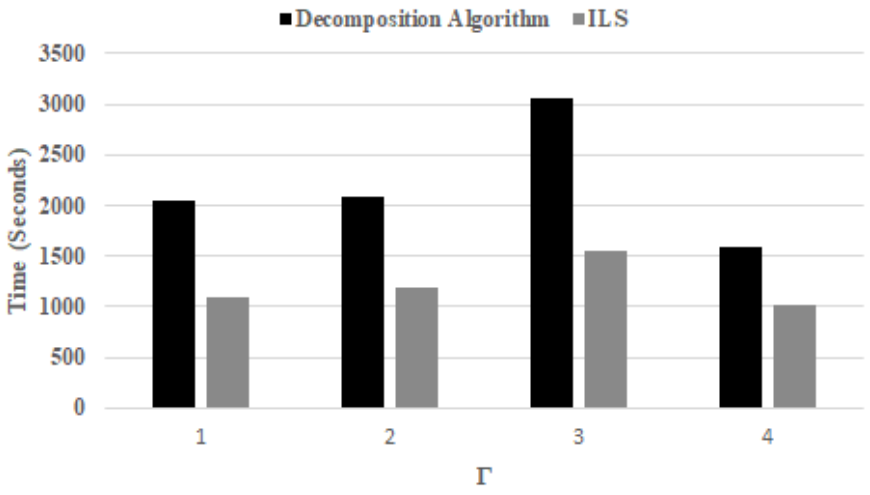

Figure 5: Comparison between the aggregated computational time required by the decompositon algorithm improved by Option 2 and by ILS heuristic to obtain the robust solution for all the instances, except for instances G1, G2 and G3, for the different values of Γ

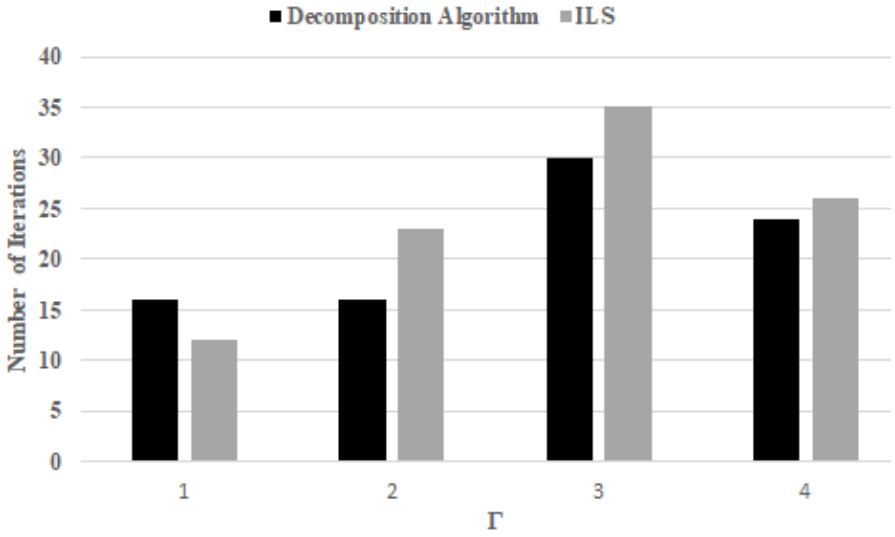

Figure 6: Comparison between the aggregated number of iterations required by the decompositon algorithm improved by Option 2 and by ILS heuristic to obtain the robust solution for all the instances, except for instances G1, G2 and G3, for the different values of Γ

For the instances G1, G2 and G3 the computational time required to find the robust solution vary a lot from strategy to strategy. The complete results for those three

instances are displayed in Table 3. The instance and the Γ value are indicated in the first and second columns, respectively. Columns "Max", "$\bar{T}$" and "Min" display, respectively, the maximum, the average and the minimum computational time used to obtain the robust solution over the nine different parameter settings. Column "DT" displays the computational time required by the decomposition algorithm improved by Option 2 to get the robust solution and in column "dif" the difference between values of columns "DT" and "$\bar{T}$" is computed. Column "T_O" reports the additional computational time used to prove the optimality of the solution obtained by the ILS heuristic. Column "$\bar{I}$" displays the average number of iterations required to find the robust solution and the column "DI" reports the number of iterations required by decomposition algorithm improved by Option 2 to get the robust solution.

Table 3: Comparison between the ILS heuristic and the decomposition algorithm for instances G1, G2 and G3.

Inst.	Γ	Max	$\bar{T}$	Min	DT	dif	T_O	$\bar{I}$	DI
G1	0	273	272	271	975	703	88	–	–
	0	305	249	176	536	287	87	–	–
	1	630	433	232	1325	892	31	2	2
G2	2	931	938	689	1960	1022	471	5	3
	3	4153	1831	1498	3010	1179	602	12	5
	4	16948	7654	6666	63734	56080	6001	16	13
G3	0	201	195	174	712	517	58	–	–
	1	829	604	466	1907	1303	527	2	4

Results not reported here show that for the easiest instances there is no advantage in using the ILS heuristic since it is necessary to spend some time in solving the box-constrained problem. When the complexity of the instances increase, the ILS heuristic becomes more efficient than the decomposition algorithm. By using the ILS heuristic, in some iterations, not all subproblems are solved to optimality, leading to worse solutions and consequently a greater number of iterations, as shown in Figure 6.

Instances G1, G2 and G3 are the most difficult instances, thus more computational time is required to obtain a solution in each iteration, and this time vary a lot from strategy to strategy. However, these are the instances that most reflect the power of the ILS heuristic since, independently of the strategy used, the corresponding computational time is always lower than the one required by the decomposition algorithm. The obtained results reveal a good performance of the ILS heuristic, since the optimal solution for each instance was always found with a small computational time.

6.3. Results for instance E3

In order to better understand the influence of the levels of protection in a robust solution we deeply analyse the computational results obtained for instance E3. Detailed information of instance E3 is given in Table 4.

Table 4: Detailed information for instance E3.

i	1	2	3	4	5
$[\underline{S}_i, \bar{S}_i]$	[0,300]	[0,350]	[0,250]	[0,145]	[0,300]
R_i	11.63	9.60	7.17	4.27	11.03
J_i	1	1	-1	-1	-1

Table 5: Cost of the robust solution for instance E3 in terms of value Γ.

Γ	0	1	2	3
Cost	226.5	230.4	248.6	267.2

In table 5, for each level of protection Γ, the cost of the robust solution is displayed. As an example of robustness, we can see that for this instance the optimal solution changes for all values of Γ. That is not the usual behavior for the remaining instances where the same solution may be optimal for more than one value of Γ. In fact we could not determine instances characteristics that allow us to identify how sensitive an instance is to the uncertainty set parameters.

For instance E3, Table 6 gives, for each value of Γ, the maximum slack time (Row *slack time*) corresponding to the maximum delay that can occur in a single arc without leading to an inventory bound violation. As expected, the slack increases as Γ increases. Row Δ gives the maximum delay that can be added to all travelling times while keeping the solution feasible. This value is lower than the value of the box constraint (which is 2) since we have the routing decisions fixed. As expected, this value also increases with the value of Γ.

Table 6: Slack time values for instance E3.

Γ	0	1	2	3
slack time	0	2	4.1	6.95
Δ	0	0.5	0.8	1.8

Figure 7 depicts the routing of an optimal solution with $\Gamma = 0$. The triple $[t_{im}, q_{imv}, s_{im}]$ is shown next to each node. We can see that $s_{41} = 0$, meaning that when the ship arrives at node 4 for the first visit the inventory level is zero, and therefore any delay in the critical path to this node will imply a shortfall. Figures 8 and 9 give two alternative optimal solutions for the case $\Gamma = 1$. The routing is the same, but the quantities loaded and unloaded and the time of visits are different. The solution in Figure 8 maximizes the minimum slack time, which is determined by the inventory level at node (5,2). The solution in Figure 8 is the one that maximizes Δ (the value considered when the travelling time in all arcs is increased by Δ), so the inventory level at nodes $(5, 2)$ and $(4, 1)$ is zero. We see from the figures that the ship travelling the dashed line route is cheaper to use than the other ship (traveling the solid line route).

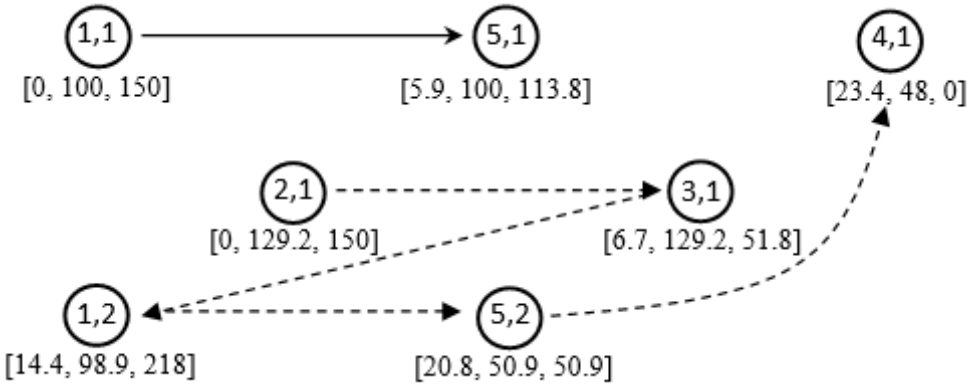

Figure 7: Optimal solution of instance E3 with $\Gamma = 0$ and cost= 226.5.

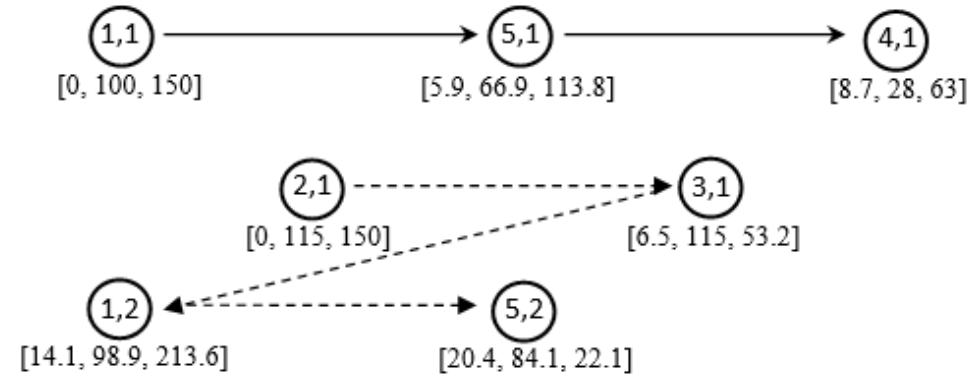

Figure 8: Optimal solution of instance E3 with $\Gamma = 1$ that maximises the minimum slack time and cost= 230.4.

Finally, Figure 10 depicts the minimum, the average and the maximum percentage of increase in cost, among all instances, as a function of Γ. Such information gives the price to obtain robust solutions. As we can see there is a wide range on the cost increase variation accordingly to the instance, from a minimum of zero to a maximum that almost doubles the cost when $\Gamma = 4$. On average, we can observe a steady increase in the cost for protection against inventory level deviations with the protection level Γ.

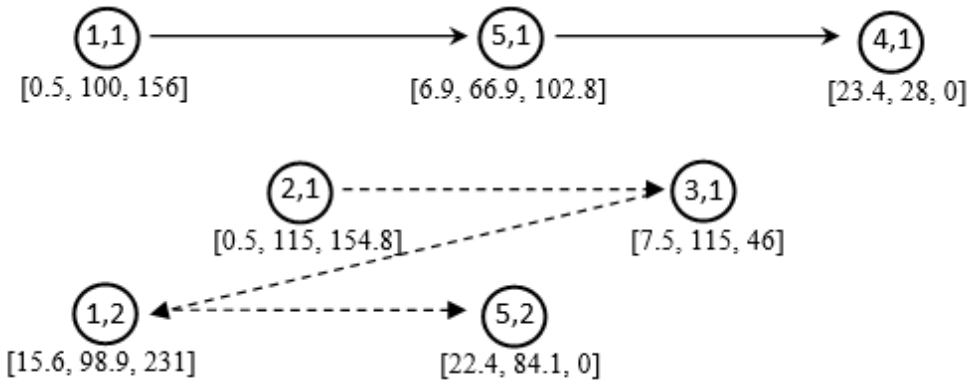

Figure 9: Routing of an optimal solution of instance E3 with $\Gamma = 1$ that maximizes Δ.

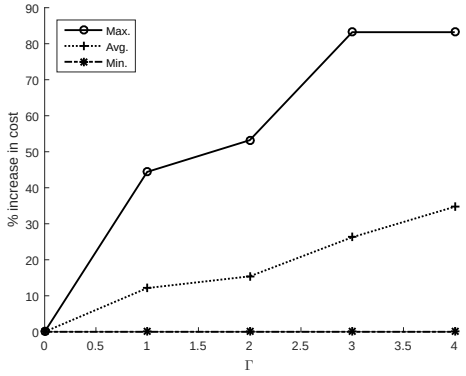

Figure 10: Percentage increase in cost as a function of Γ.

6.4. Results for large size instances

Although the original set of instances include both *easy* and *difficult* instances, in order to evaluate the performance of our approaches for larger instances, a set of 14 instances with a time horizon of 60 days, was considered. These instances were derive from those presented in Table 1 by extending the time horizon and keeping the maximum number of visits allowed to each port. Since we aim to test the most difficult cases, and the instances A4, A5, B4 and B5 are easy to solve, we omit these four cases.

In Table 7, we display the results for both the deterministic ($\Gamma = 0$) and the box-constrained problems. The instance is indicated in the first column. The last two pairs of columns give the running time in seconds (Columns *Seconds*), and the cost of the solution (Columns *Cost*).

For the box-constrained problem, instances G4 and G5 were not solved to optimality within a time limit of 18000 seconds (5 hours) and the final gap is indicated in parenthesis. This fact reinforces that these instances are difficult since both problems are deterministic

Table 7: Computational results for the two deterministic problems: $\Gamma = 0$ and box-constrained.

	Deterministic ($\Gamma = 0$)		Box-constrained	
Inst.	Seconds	Cost	Seconds	Cost
C4	9	529	23	595
C5	22	546	215	675
D4	223	379	163	414
D5	16	289	40	375
E4	39	272	870	391
E5	86	285	278	362
F4	25	387	17	486
F5	6	419	47	507
G4	2104	672	18000*	766 (20%)
G5	4194	648	18000*	775 (14%)

Table 8: Computational results for both the Decomposition Algorithm enhanced with Option 2 and ILS heuristic, for $\Gamma = 1$ and $\Gamma = 2$.

	$\Gamma = 1$				$\Gamma = 2$			
Inst.	$Time_{DA}$	$Cost_{DA}$	$Time_{ILS}$	$Cost_{ILS}$	$Time_{DA}$	$Cost_{DA}$	$Time_{ILS}$	$Cost_{ILS}$
C4	114	564	62	564	507	595	233	595
C5	838	602	483	602	838	602	486	602
D4	617	379	159	379	1522	414	1159	414
D5	137	327	97	327	141	327	147	327 (375)
E4	286	289	108	289	7702	369	6512	369
E5	412	289	112	289 (305)	3180	331	1674	331 (362)
F4	55	387	96	387 (486)	624	486	865	486
F5	1010	490	250	490	882	507	887	507
G4	2404	771*	2571	740* (787)	3604	761*	10800	761* (787)
G5	4800	785*	4032	736* (811)	10800	771*	12244	688* (820)

problems.

In Table 8 we present the results obtained for the robust problems with $\Gamma = 1$ and $\Gamma = 2$. The instance is indicated in the first column. The last two pairs of columns give, for both the decomposition algorithm improved with Option 2 and the ILS heuristic, the running time in seconds (Columns $Time_{DA}$ and $Time_{ILS}$) and the cost of the obtained solutions (Columns $Cost_{DA}$ and $Cost_{ILS}$).

Instances from C4 to F5 were solved to optimality using both approaches, thus the cost of the obtained solutions is the same. In columns $Cost_{ILS}$, the value in parenthesis is the cost of the solution obtained at the end of step 8 in Algorithm 4. This value is presented only for those cases it differs from the optimal one. Recall that for all the instances with a time horizon of 30 days, the solution obtained at the end of step 8 was always the optimal robust solution.

Analysing the computational time required by each approach we can see that, in general, the robust solutions are obtained faster when the ILS heuristic is used, as happened in the case of the instances with a time horizon of 30 days.

Instances G4 and G5 were not solved to optimality. Note that even the deterministic box-constrained problem can not be solved to optimality within a reasonable amount of

computation time, as shown in Table 7. Hence, for both the approaches, a time limit of 600 seconds was imposed to solve the restricted master problem in each iteration. This means that the obtained solutions can be suboptimal. For these two instances, better solutions were obtained using the ILS heuristic. However, the computational time required to obtain those solutions is greater than the one required by the decomposition algorithm, since more iterations were executed.

7. Conclusions

We consider a maritime inventory routing problem where the travel times are uncertain. In order to handle practical cases where the decision maker wishes to avoid inventory shortfalls at the consumers and exceed inventory capacities at producers, a robust model with recourse is proposed. The routing decisions and the quantities to load and unload are assumed to be fixed and the time of visit to ports and the inventory levels are adjustable. A two-stage decomposition procedure that considers a master problem restricted to a small subset of scenarios and a subproblem that checks whether there are violated scenarios is presented. Several improvement strategies are introduced. In particular, since the first stage has, in general, multiple alternative optimal solutions corresponding to a single set of routing decisions, we discuss approaches to choose the most robust alternative solution in relation to a given criteria. An iterated local search heuristic based on local branching ideas is introduced that allows us to obtain good quality robust solutions for all the tested instances.

A computational study based on a set of benchmark instances shows the effectiveness of the decomposition procedure with the improvement strategies in solving the robust maritime inventory routing problem. This study also shows that, in general, protecting the solutions against possible delays makes the instances harder to solve, since the running times and the number of iterations tend to increase when the number of links that can suffer a delay increases. Some insight on the robustness of solutions is obtained by comparing robust solutions against the deterministic solution, and shows that robustness is closely related to the slack time available for each time window that can be established from the inventory levels for each port visit.

Aknowledgements

The authors thank the two anonymous reviewers for their valuable comments and helpful suggestions.

The work of the first author was funded by FCT (Fundação para a Ciência e a Tecnologia) and CIDMA (Centro de Investigação e Desenvolvimento em Matemática e Aplicações) within project UID/MAT/04106/2013. The second and third author were supported financially from the Research Council of Norway through the AXIOM-project. The work of the fourth author was funded by CIDMA and FCT under Grant PD/BD/114185/2016.

References

[1] Y. Adulyasak and P. Jaillet. Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transportation Science*, 50:608–626, 2016.

[2] Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55:141–152, 2015.

[3] E.-H. Aghezzaf. Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary. *The Journal of the Operational Research Society*, 59(8):1055–1065, 2008.

[4] Agostinho Agra, Henrik Andersson, Marielle Christiansen, and Laurence A. Wolsey. A maritime inventory routing problem: Discrete time formulations and valid inequalities. *Networks*, 64:297–314, 2013.

[5] Agostinho Agra, Marielle Christiansen, and Alexandrino Delgado. Mixed integer formulations for a short sea fuel oil distribution problem. *Transportation Science*, 47:108–124, 2013.

[6] Agostinho Agra, Marielle Christiansen, Alexandrino Delgado, and Lars Magnus Hvattum. A maritime inventory routing problem with stochastic sailing and port times. *Computers & Operations Research*, 61:18–30, 2015.

[7] Agostinho Agra, Marielle Christiansen, R. Figueiredo, Lars Magnus Hvattum, Michael Poss, and Cristina Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

[8] Agostinho Agra, Marielle Christiansen, Lars Magnus Hvattum, and Filipe Rodrigues. A MIP based local search heuristic for a stochastic maritime inventory routing problem. In A. Paias, M. Ruthmair, and S. Voß, editors, *Computational Logistics*, volume 9855 of *Lecture Notes in Computer Science*, pages 18–34. Springer International Publishing, 2016.

[9] Agostinho Agra, Marcio Costa Santos, Dritan Nace, and Michael Poss. A dynamic programming approach for a class of robust optimization problems. *SIAM Journal on Optimization*, 26(3):1799–1823, 2016.

[10] Henrik Andersson, A. Hoff, Marielle Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37:1515–1536, 2010.

[11] A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[12] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[13] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[14] D. Bertsimas, D.B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53:464–501, 2011.

[15] D. Bertsimas, S. Gupta, and J. Tay. Scalable robust and adaptive inventory routing. *Available at Optimization Online*, 2016.

[16] D. Bertsimas, D. A. Iancu, and P.A. Parrilo. A hierarchy of near-optimal policies for multistage adaptive optimization. *IEEE Transactions on Automatic Control*, 56(12):2809–2824, 2011.

[17] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

[18] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[19] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54:150–168, 2006.

[20] D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5:389–414, 2008.

[21] A. Billionnet, M.-C. Costa, and P.-L. Poirion. 2-stage robust milp with continuous recourse variables. *Discrete Applied Mathematics*, 170:21–32, 2014.

[22] Gorissen BL, Yanikoğlu I, and den Hertog D. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[23] L. Cheng and M.A. Duran. Logistics for world-wide crude oil transportation using discrete event simulation and optimal control. *Computers & Chemical Engineering*, 28:897–911, 2004.

[24] Marielle Christiansen and K. Fagerholt. Robust ship scheduling with multiple time windows. *Naval Research Logistics*, 49(6):611–625, 2002.

[25] Marielle Christiansen and K. Fagerholt. Maritime inventory routing problems. In C.A. Floudas and P.M. Pardalos, editors, *Encyclopedia of Optimization*, pages 1947–1955. Springer US, Boston, MA, 2009.

[26] Marielle Christiansen and K. Fagerholt. Ship routing and scheduling in industrial and tramp shipping. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications, 2e*, chapter 13, pages 381–408. 2014.

[27] Marielle Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2013.

[28] Marielle Christiansen and B. Nygreen. Robust inventory ship routing by column generation. In G. Desaulniers, J. Desrosiers, and M. Solomon, editors, *Column Generation*, pages 197–224. Springer, New York, 2005.

[29] Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

[30] Guy Desaulniers, Jørgen G. Rakke, and Leandro C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3):1060–1076, 2016.

[31] Delage E and Iancu DA. *Robust Multistage Decision Making*. 2015.

[32] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1–3):23–47, 2003.

[33] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[34] E.E. Halvorsen-Weare and K. Fagerholt. Robust supply vessel planning. In J. Pahl, T. Reiners, and S. Voss, editors, *Network Optimization*, pages 559–574. Springer, 2011.

[35] E.E. Halvorsen-Weare, K. Fagerholt, and M. Rönnqvist. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering*, 64:290–301, 2013.

[36] Ming Li, Zheng Wang, and Felix T.S. Chan. A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. *Transportation Research Part E: Logistics and Transportation Review*, 91(Supplement C):290 – 305, 2016.

[37] C. Liebchen, M. Lübbecke, R. Möhring, and S. Stiller. *The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications*, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[38] F. Maggioni, F. Potra, and M. Bertocchi. A scenario-based framework for supply planning under uncertainty: stochastic programming versus robust optimization approaches. *Computational Management Science*, 14(1):5–44, 2017.

[39] D.J. Papageorgiou, G.L. Nemhauser, J. Sokol, M.-S. Cheo, and A.B. Keha. MIRPLib – A library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research*, 234(2):350–366, 2014.

[40] J. Rakke, H. Andersson, M. Christiansen, and G. Desaulniers. A new formulation based on customer delivery patterns for a maritime inventory routing problem. *Transportation Science*, 42:384–401, 2015.

[41] RF Roldan, R Basagoiti, and LC Coelho. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74:14–20, 2016.

[42] RF Roldan, R Basagoiti, and LC Coelho. A survey on the inventory-routing problem with stochastic lead times and demands. *Journal of Applied Logic*, 24:15–24, 2017.

[43] Robert A. Russell. Mathematical programming heuristics for the production routing problem. *International Journal of Production Economics*, 193:40–49, 2017.

[44] H.D. Sherali and S.M. Al-Yakoob. Determining an optimal fleet mix and schedules: Part I – single source and destination. *Integer programming: theory and practice*, pages 137–166, 2006.

[45] H.D. Sherali and S.M. Al-Yakoob. Determining an optimal fleet mix and schedules: Part II – multiple sources and destinations, and the option of leasing transshipment depots. *Integer programming: theory and practice*, pages 167–194, 2006.

[46] Oğuz Solyalı, Jean-François Cordeau, and Gilbert Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

[47] Oğuz Solyalı and Haldun Süral. A multi-phase heuristic for the production routing problem. *Computers & Operations Research*, 87:114–124, 2017.

[48] A. Thiele, T. Terry, and M. Epelman. Robust linear optimization with recourse. Technical report, 2010.

[49] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

[50] C. Zhang. *Robust optimization with applications in maritime inventory routing.* PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, May 2015.

[51] C Zhang, G Nemhauser, J Sokol, MS Cheon, and A Keha. Flexible solutions to maritime inventory routing problems with delivery time windows. *Computers & Operations Research*, 89:153–162, 2018.

[52] Chengliang Zhang, George Nemhauser, Joel Sokol, MS Cheon, and Dimitri Papageorgiou. Robust inventory routing with flexible time window allocation. Optimization Online, 2015.